AMERICAN JOURNAL OF MANAGEMENT AND IOT MEDICAL COMPUTING
Peer Reviewed, Referred & Indexed Journal
E-ISSN: 3069-0110
Vol.5, No.2(2) (2026)
www.ajmimc.com

# Surgery Simulator Using 3D Graphics and Real-Time Interaction

Abinash Dixit
Student, Dept. of CSE
GIFT Autonomous, Bhubaneswar
Odisha, India

Soubhagya Jena
Student, Dept. of CSE
GIFT Autonomous, Bhubaneswar
Odisha, India

Asst. Prof. Ashish Kumar Banerjee
Asst. Prof., Dept. of CSE
GIFT Autonomous, Bhubaneswar
Odisha, India

***Abstract*—The rapid advancement of medical technology and simulation-based training has significantly transformed healthcare education. Surgical training traditionally relies on supervised practice on real patients or cadavers, which presents ethical, logistical, and financial challenges. This project introduces a Surgery Simulator, an interactive virtual environment designed to provide realistic surgical training using 3D graphics and real-time user interaction.**

**The system leverages modern web technologies such as Three.js and WebGL to create a highly immersive virtual surgical environment. It incorporates physics-based modeling, real-time rendering, and user interaction through input devices to simulate surgical procedures. The simulator allows users to perform virtual operations, manipulate instruments, and observe outcomes in a controlled, risk-free environment.**

**The system architecture integrates modules for 3D visualization, physics simulation, user input handling, and procedural logic. The simulator supports multiple surgical scenarios and provides feedback based on user actions, enhancing learning effectiveness.**

**Experimental evaluation indicates that the simulator improves user understanding of surgical procedures and hand-eye coordination. The project demonstrates the potential of simulationbased learning tools in medical education and highlights future possibilities such as integration with virtual reality and haptic feedback systems.**

*Keywords*— Surgery Simulator, 3D Graphics, WebGL, Three.js, Real-Time Interaction, Medical Simulation, Virtual Training, Web-Based Application

## I. INTRODUCTION

### *A. Background*

In modern healthcare and medical education environments, simulation-based learning systems have become highly important for improving practical training and reducing risks associated with traditional surgical education methods. Conventional surgical training depends heavily on physical laboratories, cadaver practice, and direct observation of procedures, which may involve high operational costs and limited accessibility for students.

Advancements in computer graphics and web technologies have enabled the development of realistic virtual simulation systems capable of replicating surgical environments digitally. Technologies such as WebGL and Three.js support interactive 3D rendering and real-time user interaction through web browsers. These technologies allow the creation of cost-effective and accessible simulation platforms for medical training.

Modern surgery simulators provide functionalities such as tool interaction, object manipulation, real-time rendering, collision detection, and performance evaluation. These systems improve learning efficiency by allowing repeated practice in safe virtual environments without harming real patients.

### *B. Need for Virtual Surgical Training*

The demand for virtual surgical training systems has increased significantly in recent years due to the rapid advancement of healthcare technologies and modern medical education methods. Traditional practical training environments often involve limitations related to cost, patient safety, availability of equipment, and accessibility of experienced medical professionals. Surgical trainees require repeated practice for improving precision, confidence, and procedural understanding before performing operations in real clinical environments.

Virtual surgical simulators provide safe and repeatable environments where trainees can learn complex procedures without physical risks. These systems eliminate the possibility of harming real patients during training sessions and allow students to practice multiple times until they achieve confidence and accuracy. Simulation-based learning also reduces dependency on physical cadavers and expensive laboratory infrastructures.

The increasing availability of web technologies and modern graphics engines has enabled browser-based medical simulators that are accessible from different locations using standard computing devices. Such systems support flexible learning environments and improve educational accessibility for students located in remote areas.

Modern healthcare industries are increasingly adopting digital simulation technologies for improving training quality and reducing operational costs. Therefore, the development of realistic and scalable surgery simulators has become an important research area in medical technology and computer graphics domains.

### *C. Problem Statement*

Traditional surgical training systems face several limitations related to accessibility, safety, cost, and availability of resources. Practical surgical learning often requires expensive

Received: 20-04-2026 | Accepted: 29-04-2026 | Published: 04-06-2026 | www.ajmimc.com

laboratories, professional supervision, and physical resources that may not always be available for students and trainees.

Existing simulation systems may also face limitations re-lated to hardware dependency, lack of realism, high imple-mentation cost, and limited accessibility. Some systems require expensive VR devices and specialized hardware, making them difficult to deploy widely in educational institutions.

Therefore, there is a need for a web-based, low-cost, in-teractive, and scalable surgery simulator capable of providing realistic 3D visualization and real-time interaction using mod-ern web technologies.

### D. Objectives

The major objective of the proposed Surgery Simulator system is to develop a realistic and interactive virtual surgical training platform using 3D graphics and real-time interaction technologies.

The system aims to:

- Provide realistic 3D visualization of surgical environ-ments
- Support real-time interaction with virtual surgical tools
- Improve accessibility of medical training systems
- Reduce training cost and operational risks
- Provide scalable architecture for future enhancements

### E. Scope of the Project

The proposed system focuses on creating a web-based surgi-cal simulation environment that supports interactive learning and virtual surgical practice. The simulator can be used in educational institutions, online medical training platforms, and healthcare learning environments.

The architecture also supports future enhancements such as virtual reality integration, AI-based performance analysis, multiplayer collaboration, gesture recognition, and advanced tissue simulation technologies.

## II. LITERATURE REVIEW

### A. Existing Surgical Simulation Systems

Modern surgical simulators have transformed medical edu-cation by providing safe and repeatable virtual training envi-ronments. Existing systems support realistic surgical practice, anatomy visualization, and procedural learning through inter-active simulation technologies.

Several advanced simulators use VR and AR technologies for immersive training experiences. These systems improve practical learning and reduce risks associated with real surgical procedures. However, many existing systems require expensive hardware and specialized devices, limiting accessibility for smaller institutions.

### B. Modern 3D Graphics Technologies

Recent advancements in 3D graphics technologies have enabled the development of realistic and interactive web-based simulations. Technologies such as WebGL and Three.js sup-port hardware-accelerated rendering, lighting effects, texture mapping, and real-time scene management.

Three.js simplifies complex rendering operations and provides high-level abstractions for creating 3D environ-ments. WebGL utilizes GPU acceleration for rendering high-performance graphics directly within web browsers.

### C. Applications of 3D Graphics in Healthcare

Three-dimensional graphics technologies are widely used in modern healthcare applications for improving visualization, medical analysis, simulation, and training systems. Advanced graphics rendering allows medical professionals and students to understand anatomical structures, surgical procedures, and internal body systems more effectively through interactive visual environments.

Medical imaging systems such as CT scans, MRI scans, and ultrasound technologies frequently utilize 3D reconstruction techniques for improving diagnosis and treatment planning. Interactive 3D visualization helps surgeons analyze patient anatomy before performing complex operations. These tech-nologies also improve communication between doctors and patients by providing clearer medical explanations.

Virtual surgery simulators use 3D graphics to replicate real-istic operating environments and surgical instruments. Light-ing effects, texture mapping, shading models, and animation techniques improve visual realism and user immersion within the simulation environment. Real-time rendering also enables smooth interaction between users and virtual objects during training sessions.

The integration of 3D graphics with artificial intelligence, virtual reality, and augmented reality technologies is further improving medical education and digital healthcare systems. These advancements support more accurate simulation en-vironments and improve practical learning experiences in modern healthcare industries.

### D. Importance of Simulation-Based Medical Learning

Simulation-based medical learning improves understand-ing of anatomy, surgical procedures, and practical decision-making. Students can repeatedly practice operations in safe virtual environments without physical risks.

Virtual simulation systems also reduce operational costs and improve accessibility for remote learning environments. These technologies provide performance feedback and inter-active guidance mechanisms that improve medical learning efficiency.

### E. Research Gap

Although many surgical simulators provide realistic training environments, several systems still face limitations related to accessibility, hardware dependency, scalability, and implemen-tation cost.

Many advanced simulators require dedicated VR hardware and expensive installations. Existing systems may also lack browser-based accessibility, responsive interaction, and scal-able architecture.

Therefore, there is a need for a low-cost, browser-based surgical simulator capable of providing real-time interaction and realistic 3D visualization using modern web technologies.

## III. SYSTEM OVERVIEW

### A. Proposed System

The proposed Surgery Simulator is designed as a web-based interactive platform for simulating surgical procedures using 3D graphics and real-time interaction technologies.

The system allows users to interact with virtual surgical tools and anatomical structures within a simulated operation theatre environment. Real-time rendering, collision detection, and feedback mechanisms improve interaction quality and learning efficiency.

### B. System Architecture

The proposed system follows a layered architecture consisting of:

- Input Layer
- Control and Logic Layer
- Simulation Layer
- Rendering Layer
- Feedback Layer

Each layer performs specific functionalities and communicates with adjacent layers for maintaining smooth system operation and real-time responsiveness.

### C. Advantages of Web-Based Architecture

The web-based architecture improves accessibility and portability by allowing users to access the simulator directly through browsers without additional installations.

The architecture also simplifies deployment, maintenance, and scalability. Cross-platform compatibility enables the simulator to run on desktops, laptops, and mobile devices efficiently.

### D. Importance of Real-Time Interaction

Real-time interaction is one of the most important components of modern simulation systems because it allows users to perform actions and receive immediate visual responses from the system. In surgical training environments, real-time responsiveness improves user immersion and creates realistic operational experiences for trainees.

The proposed Surgery Simulator continuously processes user inputs such as mouse movements, clicks, and keyboard commands while updating the simulation environment dynamically. Real-time communication between the rendering engine, simulation engine, and input system ensures smooth interaction with virtual surgical tools and anatomical structures.

Low-latency interaction mechanisms are necessary for maintaining realism during simulation sessions. Delays in system response may negatively affect training quality and reduce the effectiveness of procedural learning. Therefore, optimized rendering pipelines and efficient event handling mechanisms are implemented within the proposed system.

Real-time interaction technologies also improve decision-making skills and hand-eye coordination among users. These features make simulation-based medical learning more practical, effective, and engaging for trainees and educators.

### E. Key Features

The proposed system provides several important features:

- Real-time interaction
- 3D visualization
- Collision detection
- Surgical tool simulation
- Responsive rendering
- Feedback generation
- Cross-platform accessibility
- Modular architecture

### F. User Modules

The system mainly consists of trainee and administrator modules. Trainees can perform virtual surgical procedures and interact with simulation tools, while administrators can monitor performance and manage training environments.

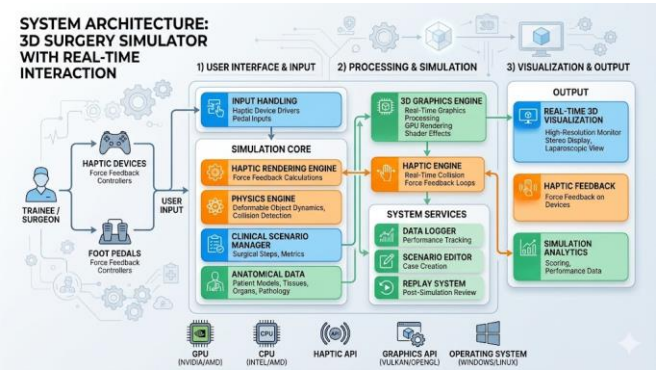


Fig. 1. System Architecture Diagram

## IV. METHODOLOGY

### A. Workflow of the System

The workflow begins when users select a surgical procedure from the available options. The simulator then loads corresponding 3D models, tools, and environments.

Users interact with virtual surgical tools using input devices such as mouse and keyboard. The simulation engine processes user actions and updates object states dynamically. The rendering engine continuously updates the scene and provides visual feedback to the user.

### B. Input Handling Methodology

The Input Module captures mouse movements, keyboard inputs, clicks, and drag actions for enabling interaction with virtual tools and objects.

Event listeners process user interactions and convert them into system commands. Accurate input mapping ensures precise control and smooth user experience during simulation.

### C. Simulation Engine Methodology

The Simulation Engine manages object interactions, collision detection, and real-time updates of surgical environments. The engine continuously processes user actions and applies simulation logic to maintain realistic system behavior. Object properties such as position, orientation, and interaction states are dynamically updated during execution.

### D. Rendering Methodology

The Rendering Module uses Three.js and WebGL for generating realistic 3D scenes and animations.

The rendering pipeline includes:

- Scene creation
- Object placement
- Lighting setup
- Camera configuration
- Real-time frame rendering

### E. Feedback Generation Methodology

The system provides immediate feedback through visual cues, color changes, and performance metrics.

Feedback mechanisms improve learning efficiency by helping users identify mistakes and improve procedural accuracy during training sessions.

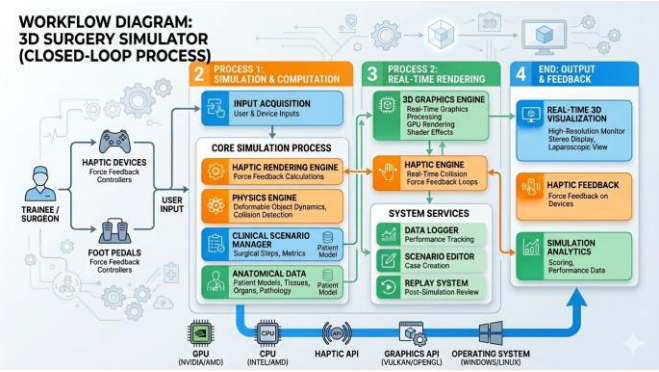


Fig. 2. Workflow Diagram

## V. SYSTEM DESIGN

### A. Overview of System Design

The system design of the proposed Surgery Simulator Using 3D Graphics and Real-Time Interaction focuses on developing a modular, scalable, and interactive architecture capable of supporting realistic surgical simulation within a web-based environment. The design integrates multiple software components such as rendering systems, input processing modules, simulation engines, collision detection mechanisms, and feedback systems into a unified platform.

The primary objective of the system design is to ensure smooth communication between graphical rendering components and real-time interaction modules while maintaining high performance and user responsiveness. The architecture is designed to support future scalability, maintainability, and advanced feature integration such as virtual reality support, AI-based analysis, and multiplayer simulation environments.

The proposed design follows modular software engineering principles where each module performs dedicated functional-ities independently while communicating with other modules through structured data flow mechanisms. This approach im-proves system organization, debugging efficiency, scalability, and future extensibility.

The overall architecture mainly consists of:

- User Interface Module
- Input Handling Module
- Rendering Engine
- Simulation Engine
- Collision Detection Module
- Animation Module
- Feedback and Evaluation Module
- Data Management Module

Each module contributes to maintaining realistic simulation behavior and interactive user experience within the virtual surgical environment.

### B. System Architecture Design

The proposed system follows a layered architecture where each layer performs specialized operations related to ren-dering, simulation, interaction, and system management. The layered design improves modularity and simplifies communi-cation between different application components.

The architecture consists of the following layers:

1) Presentation Layer
2) Interaction Layer
3) Simulation Processing Layer
4) Rendering Layer
5) Data Management Layer

The Presentation Layer provides graphical user interfaces and displays simulation environments to users. This layer includes menus, simulation controls, tool panels, and visual-ization windows.

The Interaction Layer processes user inputs received through mouse and keyboard devices. It interprets user actions and forwards interaction events to the simulation engine for further processing.

The Simulation Processing Layer manages simulation logic, collision detection, procedural workflows, and object inter-action handling. It acts as the core operational layer of the system.

The Rendering Layer handles graphical rendering opera-tions using WebGL and Three.js technologies. It continuously updates the visual environment and maintains real-time ren-dering performance.

The Data Management Layer manages simulation data, ob-ject properties, interaction records, and configuration settings. It supports efficient data storage and retrieval operations during simulation execution.

The layered architecture improves separation of concerns and enables independent optimization of different system components.

### C. User Interface Design

The User Interface (UI) is designed to provide simple, re-sponsive, and user-friendly interaction mechanisms for trainees and administrators. The interface layout is organized carefully to minimize operational complexity while maintaining efficient accessibility to simulation controls and functionalities.

The UI mainly consists of:

- Navigation Menu
- Simulation Window
- Tool Selection Panel
- Control Buttons
- Feedback Display Area
- Performance Indicators
- Camera Control Options

The Navigation Menu allows users to access simulation modules, settings, and training options. The Simulation Win-dow displays the 3D surgical environment where users interact with virtual objects and tools.

The Tool Selection Panel enables users to select different surgical instruments and operational modes during training sessions. Control Buttons are used for starting, pausing, re-setting, and managing simulation activities.

Feedback Display Areas provide real-time notifications, procedural guidance, and error alerts during operation. Per-formance indicators display simulation statistics such as com-pletion time, interaction accuracy, and procedural efficiency.

Responsive design techniques are implemented to ensure compatibility across different screen resolutions and devices. CSS styling and dynamic layouts improve visual appearance and usability within the application environment.

### D. Input Handling Module Design

The Input Handling Module is responsible for capturing user interactions and converting them into simulation commands. This module continuously monitors mouse and keyboard events during simulation execution.

The input system supports:

- Mouse movement tracking
- Object selection
- Drag-and-drop interaction
- Camera navigation
- Keyboard-based controls
- Simulation command handling

Event listeners are implemented for detecting user actions such as clicks, mouse dragging, key presses, and scrolling operations. These events are processed and forwarded to the simulation engine for generating corresponding responses.

Raycasting mechanisms are used for object selection within the 3D environment. Invisible rays are projected from the camera toward cursor positions for identifying intersecting objects and enabling precise interaction handling.

The Input Handling Module is optimized for maintaining low interaction latency and smooth operational responsiveness during real-time simulation activities.

## VI. IMPLEMENTATION

### A. Frontend Implementation

The frontend implementation of the proposed Surgery Sim-ulator Using 3D Graphics and Real-Time Interaction is de-veloped using HTML, CSS, JavaScript, and Three.js tech-nologies. The frontend layer is responsible for providing interactive user interfaces, rendering 3D objects, managing user interaction, and displaying real-time simulation feedback within the browser environment.

The frontend architecture follows a modular design ap-proach where different interface components are organized separately for improving maintainability and scalability. User interface elements such as menus, control panels, simulation windows, tool selectors, and feedback indicators are integrated into a responsive layout that supports smooth navigation and interaction.

HTML is used for structuring the simulation environment and interface components, while CSS is used for styling, responsive layouts, animations, and visual customization. JavaScript manages event handling, rendering logic, object manipulation, and communication between different frontend modules.

Responsive frontend design techniques are implemented to support compatibility across desktops, laptops, and tablets. The interface dynamically adjusts according to screen size and device resolution to improve accessibility and usability for users operating on different platforms.

The frontend also supports interactive functionalities such as:

- Camera movement and scene navigation
- Surgical tool selection
- Object dragging and manipulation
- Real-time feedback display
- Simulation control operations
- Performance monitoring
- Dynamic object interaction

The implementation focuses on minimizing interface com-plexity while maintaining realistic interaction capabilities for improving medical learning experiences.

### B. Three.js Integration

Three.js is one of the core technologies used in the imple-mentation of the proposed surgery simulator. It is a JavaScript-based 3D graphics library that simplifies WebGL programming and supports efficient rendering of interactive 3D scenes directly within web browsers.

The integration of Three.js enables the creation of realistic surgical environments consisting of:

- 3D anatomical structures
- Surgical instruments
- Operation theatre environments
- Interactive simulation objects
- Lighting and shadow systems
- Real-time animations

The rendering process begins with scene creation, where all graphical objects and environmental components are added into a centralized 3D scene structure. Cameras are configured for observing the scene from different angles, while rendering engines continuously update graphical output during simula-tion execution.

Three.js provides several built-in functionalities that sim-plify implementation processes such as:

- Geometry generation
- Material management
- Texture mapping
- Lighting configuration
- Animation handling
- Camera controls
- Object transformations

The system uses PerspectiveCamera for realistic depth visualization and WebGLRenderer for hardware-accelerated rendering. Different mesh objects are created for representing surgical instruments and anatomical structures. Texture mate-rials are applied to improve realism and visual clarity during simulation sessions.

Three.js also supports advanced graphical operations such as transparency handling, shadow generation, environmental reflections, and smooth object transformations. These capabil-ities improve visual realism and user immersion within the simulation environment.

### C. 3D Model Loading and Management

The proposed Surgery Simulator uses multiple 3D models representing surgical tools, anatomical structures, operating ta-bles, and simulation environments. These models are imported into the application using standard formats such as:

- GLTF
- GLB
- OBJ
- FBX

Three.js loaders such as GLTFLoader and OBJLoader are used for importing and processing 3D assets efficiently. Im-ported models are scaled, rotated, textured, and positioned dy-namically within the simulation scene according to operational requirements.

Model optimization techniques are implemented to reduce polygon count and improve rendering performance. Texture compression and efficient material management reduce mem-ory usage and improve graphical responsiveness during simu-lation sessions.

Hierarchical scene organization is also implemented for managing multiple simulation objects efficiently. Grouping mechanisms simplify object transformations and improve maintainability of complex 3D environments.

The system also supports dynamic model interaction where users can manipulate surgical tools and interact with anatom-ical structures during simulation activities. Smooth animation transitions and object synchronization mechanisms improve realism and operational continuity.

### D. Scene Management and Lighting

Scene management is responsible for organizing graphical objects, environmental components, cameras, and lighting sys-tems within the virtual simulation environment. Efficient scene organization improves rendering performance and simplifies interaction management during real-time operation.

Different types of lighting models are implemented within the proposed system including:

- Ambient Lighting
- Directional Lighting
- Point Lighting
- Hemisphere Lighting

Ambient lighting provides general scene illumination while directional lighting simulates focused surgical lighting condi-tions. Point lighting is used for localized illumination effects around tools and operating regions.

Shadows and shading effects improve depth perception and enhance realism within the virtual operating environment. Material properties such as roughness, metalness, and reflec-tivity are adjusted for improving the appearance of surgical instruments and anatomical structures.

Camera positioning mechanisms allow users to observe the simulation from multiple viewing angles. OrbitControls are implemented for enabling camera rotation, zooming, and smooth navigation around the surgical environment.

Scene optimization techniques such as frustum culling, object grouping, and texture optimization are implemented to improve rendering efficiency and maintain stable performance during complex simulation operations.

### E. ER Diagram

The Entity Relationship (ER) Diagram represents the database structure and relationships between different entities used within the proposed Surgery Simulator system. The ER model helps in organizing system data efficiently and improves understanding of how different modules interact with each other during simulation operations.

The proposed system mainly consists of entities such as User, Simulation Session, Surgical Tool, Performance Report, Feedback Module, and Simulation Environment. Each entity stores important information related to user interaction, sim-ulation activities, evaluation records, and operational configu-rations.

The User entity stores information related to trainees and administrators such as user ID, username, login credentials, training records, and performance history. This entity helps in maintaining individual user profiles and monitoring simulation activities.

The Simulation Session entity manages details about on-going and completed simulation activities. It stores session IDs, simulation timestamps, selected procedures, operational duration, and interaction statistics. This entity helps in tracking simulation progress and analyzing user performance.

The Simulation Environment entity stores information re-lated to virtual scenes, anatomical models, lighting configu-

AMERICAN JOURNAL OF MANAGEMENT AND IOT MEDICAL COMPUTING
Peer Reviewed, Referred & Indexed Journal
E-ISSN: 3069-0110
Vol.5, No.2(2) (2026)
www.ajmimc.com

rations, and graphical settings. It helps maintain consistency and organization within the rendering environment.

Relationships between entities are established using unique identifiers and references. A single user can participate in multiple simulation sessions, while each simulation session may involve multiple surgical tools and performance evaluations. The ER design improves data organization, maintainability, and future scalability within the proposed system.

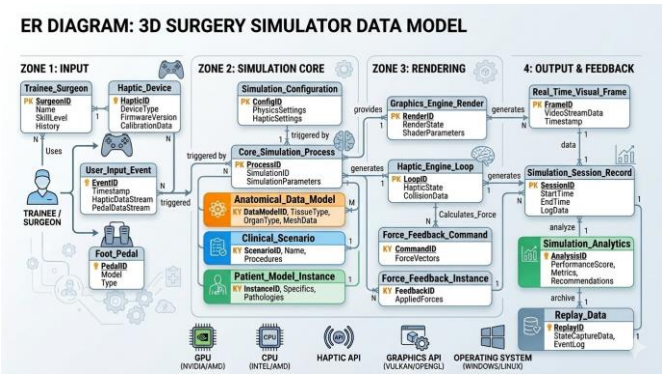


Fig. 3. Entity Relationship Diagram

*F. Data Flow Diagram*

The Data Flow Diagram (DFD) represents the movement of data and communication flow between different modules of the proposed Surgery Simulator system. It explains how information is processed, transferred, and updated during simulation execution and user interaction.

The DFD begins with the User entity, which interacts with the system through mouse and keyboard inputs. User actions such as tool selection, object manipulation, camera movement, and procedural operations are captured by the Input Handling Module.

The Input Handling Module processes user commands and forwards interaction data to the Simulation Engine. The Simulation Engine acts as the core processing unit responsible for managing simulation logic, object interaction, collision handling, animation synchronization, and procedural workflows.

The Collision Detection Module continuously monitors object intersections and interaction boundaries within the simulation environment. When collisions occur between surgical tools and anatomical structures, collision data is transmitted back to the Simulation Engine for further processing.

The Rendering Engine receives updated scene information from the Simulation Engine and generates graphical outputs using Three.js and WebGL technologies. Real-time rendering continuously updates object positions, animations, lighting effects, and scene visualization during operation.

The Feedback and Evaluation Module analyzes user interactions and simulation events to generate performance statistics, procedural guidance, and error notifications. These feedback mechanisms are displayed to users dynamically within the simulation environment.

The Data Management Module stores simulation records, user performance statistics, interaction history, and configuration settings for future analysis and evaluation purposes.

The DFD improves understanding of communication flow and coordination between different software components. It also helps developers identify data dependencies, processing sequences, and interaction mechanisms within the system architecture.

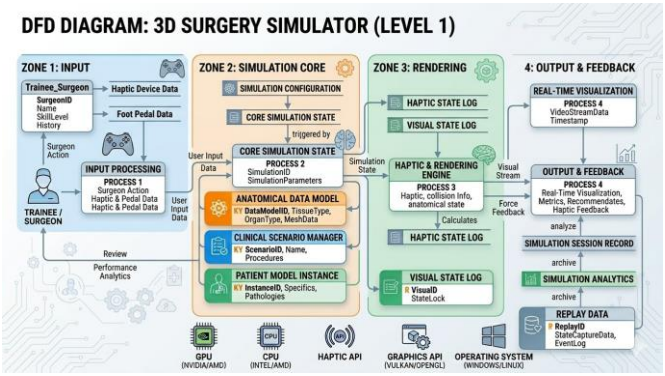


Fig. 4. Data Flow Diagram

Frustum culling prevents rendering of objects outside the camera view, reducing GPU workload and improving rendering efficiency. Efficient object grouping and batching further optimize rendering operations during complex scene execution.

Performance monitoring tools are also integrated for analyzing frame rate, GPU utilization, and rendering stability during simulation sessions. These optimization techniques improve scalability and ensure compatibility across systems with different hardware capabilities.

## VII. RESULTS AND ANALYSIS

*A. Performance Evaluation*

The system was tested on different hardware configurations for evaluating responsiveness and rendering performance.

*B. Frame Rate Analysis*

The simulator maintained stable rendering performance:

- High-End Systems: 55–60 FPS
- Mid-Range Systems: 40–50 FPS
- Low-End Systems: 25–35 FPS

*C. Simulation Accuracy*

The system achieved efficient interaction handling and collision detection accuracy during testing.

*D. Usability Analysis*

Users adapted quickly to the simulator interface. The controls were intuitive and required minimal training.

*E. Visualization Analysis*

The simulator achieved realistic visualization through lighting, shading, and smooth animation effects.

| Parameter | Traditional Training | Simulator System |
|---|---|---|
| Cost | High | Low |
| Risk | High | None |
| Accessibility | Limited | High |
| Repeatability | Limited | Unlimited |
| Practical Exposure | Moderate | Interactive |
| Scalability | Low | High |

Table 1. Comparative Analysis

*F. Comparative Analysis*

## VIII. DISCUSSIONS

*A. Advantages of the System*

The proposed system provides several advantages including low-cost training, real-time interaction, responsive visualization, and cross-platform accessibility.

*B. Limitations*

The simulator currently lacks advanced haptic feedback and realistic soft tissue deformation mechanisms.

*C. Security and Reliability*

The modular architecture improves reliability and prevents major system failures during operation.

*D. Real-World Applicability*

The simulator can be used effectively in medical colleges, online healthcare education platforms, and remote surgical training systems.

*E. Scalability and Future Improvements*

The architecture supports future integration of VR, AR, AI guidance, multiplayer collaboration, and cloud-based deployment systems.

## IX. FUTURE SCOPE

*A. Virtual Reality Integration*

Future versions may integrate VR headsets for immersive training environments.

*B. Haptic Feedback*

Advanced tactile feedback devices may improve realism and practical learning experiences.

*C. AI-Based Guidance*

AI technologies can provide automated feedback, voice-based guidance, and intelligent performance evaluation systems.

*D. Advanced Physics Engine*

Future systems may simulate realistic tissue deformation and advanced surgical interactions.

*E. Multi-User Collaboration*

The simulator may support remote collaborative surgical training with multiple participants.

*F. Cloud Deployment*

Cloud-based infrastructure may improve scalability, accessibility, and centralized management.

## X. CONCLUSION

The proposed Surgery Simulator Using 3D Graphics and Real-Time Interaction successfully demonstrates the application of modern web technologies and real-time graphics for medical training.

*A. Collision Detection Technique*

Collision detection is an important component of the proposed Surgery Simulator because it enables accurate interaction between virtual surgical tools and anatomical structures. The collision detection mechanism continuously monitors the positions and movements of objects within the simulation environment.

Bounding box algorithms and raycasting techniques are used for identifying intersections between tools and target objects. When collisions occur, the system updates interaction states and generates corresponding visual responses. These mechanisms improve simulation realism and maintain procedural accuracy during virtual surgical operations.

Efficient collision handling is necessary for maintaining smooth system performance and preventing incorrect object interactions. Optimized mathematical calculations and spatial partitioning techniques reduce computational overhead and improve real-time responsiveness.

The collision detection methodology also supports future enhancements such as soft tissue interaction, physics-based deformation, and advanced haptic feedback integration for improving realism in medical training environments.

The system integrates 3D visualization, real-time interaction, collision detection, and feedback mechanisms into a single web-based platform. The implementation using Three.js, WebGL, and JavaScript improves accessibility, scalability, and performance.

The simulator provides safe and repeatable surgical training environments while reducing operational costs and risks associated with traditional medical learning systems.

Overall, the proposed system establishes a strong foundation for future advancements in virtual medical education and interactive healthcare simulation technologies.

## REFERENCES

[1] N. E. Seymour et al., “Virtual reality training improves operating room performance,” Annals of Surgery, 2002.
[2] R. Aggarwal and A. Darzi, “Simulation to enhance patient safety,” BMJ, 2011.
[3] R. Szeliski, “Computer Vision: Algorithms and Applications,” Springer, 2010.
[4] W. R. Sherman and A. B. Craig, “Understanding Virtual Reality,” Morgan Kaufmann, 2003.
[5] Three.js Documentation, 2024.
[6] OpenGL Architecture Review Board, “OpenGL Programming Guide,” 2017.
[7] Google MediaPipe Documentation, 2023.
[8] J. Zhang et al., “Artificial intelligence in surgical training and assessment,” Journal of Surgical Education, 2018.

Received: 20-04-2026 | Accepted: 29-04-2026 | Published: 04-06-2026 | www.ajmimc.com